

REMARKS

Claims 1-9, and 11-20 are pending and under consideration. Claim 1 is amended herein. Support for the amendment appears in Fig. 9, as well as at page 14, lines 5-8 of the specification. Further reconsideration is requested based on the foregoing amendment and the following remarks.

Advisory Action Mailed June 22, 2007:

The Applicants again appreciate the consideration given to their arguments. The Applicants, however, are disappointed that their arguments were not found to be persuasive.

The continuation sheet attached to the Advisory Action asserts that lines 4-7:

The term used in the claims is a first task, a second task, and a third task. There is no clear understanding of how first, second, and third regarding a task illustrates the difference in the tasks. The interpretation of first task, second task, third task is unlimiting, some of which are provided below. One interpretation is that the first, second, and third task are subtasks of an overall task, such that the switching performs switching between the different subtask.

The claims, to the contrary, recite first, second, and third tasks, not subtasks of an overall task, as noted in the Advisory Action. Thus, to interpret first, second, and third, tasks as subtasks of an overall task is submitted to be without basis.

The continuation sheet attached to the Advisory Action asserts further at lines 7 and 8:

Another interpretation is a combination of the tasks are on part of the same task.

This interpretation is also submitted to be without basis since, as noted in the Advisory Action, the claims recite first, second, and third tasks, not a combination of the tasks that are part of the same task.

The continuation sheet attached to the Advisory Action asserts further at lines 15 and 16, that:

Applicant's argument that one of the calling or called routines is the ISR is improper, because the ISR is in the supervisor mode, e.g. lower protection domain, from the user tasks.

The applicants asserted, rather, that an exception in Fleck, such as an external interrupt or a trap, is handled like a call instruction. In particular, as described at column 3, lines 49-53:

In case of a call instruction, an exception such as an external interrupt or a trap, or a system call instruction, the context switch control unit 3c automatically stores the context into a memory field, the so-called context save areas.

Fleck, moreover, only says that the software managed tasks are *assumed* to execute in user mode, whereas ISR are *expected* to execute in supervisor mode. Nowhere does Fleck say that ISR *have* to execute in supervisor mode. In particular, as described at column 3, lines 14-19:

Software managed tasks are sometimes referred to elsewhere as user tasks. That reflects an assumption that they will execute in user mode, whereas ISR are expected to execute in supervisor mode.

And in embedded system software managed tasks may be running in supervisor mode as well. In particular, as described at column 3, lines 19 and 20:

In embedded systems, however, SMTs often run in supervisor mode, for efficient access to system resources.

It is thus just as likely, based on the disclosure of Fleck, that in the event of an exception occurring during execution of a called task, the processor would switch *back* to the calling task.

The continuation sheet attached to the Advisory Action asserts further at lines 17 and 18, that:

Applicant remaining arguments apply that Fleck either singly or in combination does not teach returning execution to a previous task, i.e. second, third, etc.

Applicants, to the contrary, argued that Fleck neither teaches, discloses, nor suggests "the task switching function executing a return operation to the third data processing task," which is different from a first or a second data processing task. Still, in the interest of compact prosecution only, and not for any reason of patentability, claim 1 has been amended further to recite:

The task switching function selecting a return address corresponding to a second data processing task different from the first data processing task.

And:

The second data processing task executing a call to the task switching function; the task switching function selecting a return address corresponding to a third data processing task different from the first and the second data processing tasks.

The continuation sheet attached to the Advisory Action asserts further at lines 24 and 25:

To exit a called function or trap or interrupt handler and switch back to the previous context.

This is to be contrasted with the claimed invention, in which "the task switching function executing a return operation to the third data processing task," which is different from a first or a second data processing task.

The continuation sheet attached to the Advisory Action asserts further, at lines 28, 29, and 30:

From at least the above recitations, Fleck teaches returning execution to a previous task whose context was saved, by executing a return operation that takes the head of the previous context list and loads its values.

This is to be contrasted with the claimed invention, in which "the task switching function executing a return operation to the third data processing task," which is different from a first or a second data processing task.

The continuation sheet attached to the Advisory Action asserts further at line 30, that:

By the amount of times this is executed, there are different values.

This is submitted to have no basis in Fleck. If the values had changed since the context was saved, it would not be worth returning to that context.

The final Office Action mailed March 8, 2007:

The final Office Action asserts at page 12, lines 7-10, that:

Applicant states that the third task is a different task and then the second data processing task which in turn is a different task than the first data processing task. The examiner disagrees. There is no express language in the claims that indicates that any of the task are different or in what way they are different from one another.

This is submitted to be incorrect. To the contrary, the use of the recitation "a second," in the second clause of claim 1 indicates that "a second data processing task" is different than "a first data processing task" recited in the first clause of claim 1. If, on the other hand, "a second data processing task" were the same data processing task as "a first data processing task," then the second clause of claim 1 would have recited "the first data processing task" instead of "a second data processing task."

Similarly, the use of the recitation "a third" in the sixth clause of claim 1 indicates that "a third data processing task" is different than either "a first data processing task" recited in the first clause of claim 1 or "a second data processing task" recited in the second clause of claim 1. If, on the other hand, "a third data processing task" were the same data processing task as "a first data processing task," for example, then the sixth clause of claim 1 would have recited "the first data processing task" instead of "a third data processing task."

The final Office Action asserts at page 12, lines 10-16, that:

Fleck teaches switching execution context of a processor from a calling task to a called task and switching context from executing task to an exception handler. Fleck teachings inherently disclose that a calling task switches context to a called task such that when the called task executes and an error occurs, it switches context to an exception handler.

This is submitted to be incorrect. There is no disclosure in Fleck, inherently or otherwise, of "a calling task switches context to a called task such that when the called task executes and an error occurs, it switches context to an exception handler," contrary to the assertion in the final Office Action. In Fleck, rather, an exception, such as an external interrupt or a trap, is handled like a call instruction. In particular, as described at column 3, lines 49-53:

In case of a call instruction, an exception such as an external interrupt or a trap, or a system call instruction, the context switch control unit 3c automatically stores the context into a memory field, the so-called context save areas.

It is thus just as likely, based on the disclosure of Fleck, that in the event of an exception occurring during execution of a called task, the processor would switch *back* to the calling task.

Furthermore, as provided in M.P.E.P. § 2112(IV):

The fact that a certain result or characteristic may occur or be present in the prior art is not sufficient to establish the inherency of that result or characteristic. *In re Rijckaert*, 9 F.3d 1531, 1534, 28 USPQ2d 1955, 1957 (Fed. Cir. 1993).

Thus, even if it were true that "a calling task switches context to a called task such that when the called task executes and an error occurs, it switches context to an exception handler," *may* occur in Fleck, that would still not be sufficient to establish the inherency of that result or characteristic, *In re Rijckaert*.

Moreover,

"To establish inherency, the extrinsic evidence 'must make clear that the missing descriptive matter is necessarily present in the thing described in the reference, and that it would be so recognized by persons of ordinary skill. Inherency, however, may not be established by probabilities or possibilities. The mere fact that a certain thing may result from a given set of circumstances is not sufficient.' " *In re Robertson*, 169 F.3d 743, 745, 49 USPQ2d 1949, 1950-51 (Fed. Cir. 1999).

Thus, even if it were true that "a calling task switches context to a called task such that when the called task executes and an error occurs, it switches context to an exception handler," *may* result from a given set of circumstances in Fleck, that would still not be sufficient to establish inherency, *In re Robertson*.

Finally:

"In relying upon the theory of inherency, the examiner must provide a basis in fact and/or technical reasoning to reasonably support the determination that the allegedly inherent characteristic necessarily flows from the teachings of the applied prior art." *Ex parte Levy*, 17 USPQ2d 1461, 1464 (Bd. Pat. App. & Inter. 1990).

Since the determination that "a calling task switches context to a called task such that when the called task executes and an error occurs, it switches context to an exception handler," does not necessarily flow from the teachings of Fleck, reliance on the theory of inherency is submitted to be misplaced, *Ex parte Levy*. Further reconsideration is thus requested.

Rejection Under 35 U.S.C. § 102:

Claims 1-8, 11-17, and 19 were rejected under 35 U.S.C. § 102(b) as anticipated by Fleck *et al.*, US 6,128,641 (hereinafter "Fleck"). The rejection is traversed to the extent it might apply to the claims as amended. Reconsideration of the rejection is respectfully requested.

Claim 1 recites:

The task switching function executing a return operation to the third data processing task.

Fleck neither teaches, discloses, nor suggests "the task switching function executing a return operation to the third data processing task," as recited in claim 1. The final Office Action does not even assert specifically that Fleck does teach a task switching function "executing a return operation to the third data processing task," as recited in claim 1. Furthermore, even though Fleck shows three context save areas in Fig. 4, only the context save area in the front of the previous context list ever gets written to. Since, in Fleck, the processor context is simply saved to whichever context save area has been transferred to the front of the previous context list, which in this case is the CSA3, Fleck is not "executing a return operation to the third data processing task," as recited in claim 1. Claim 1 is thus submitted to be allowable. Withdrawal of the rejection of claim 1 is earnestly solicited.

Claims 2-8 depend from claim 1 and add further distinguishing elements. Claims 2-8 are thus also submitted to be allowable. Withdrawal of the rejection of claims 2-8 is earnestly solicited.

Claims 11-16:

Claim 11 recites:

The memory management apparatus responsive to said instruction information

indicating a return instruction for moving said return address corresponding to said third task from said second storage location to a register of the data processor

Fleck neither teaches, discloses, nor suggests "the memory management apparatus responsive to said instruction information indicating a return instruction for moving said return address corresponding to said third task from said second storage location to a register of the data processor," as discussed above with respect to the rejection of claim 1. Claim 11 is thus submitted to be allowable. Withdrawal of the rejection of claim 11 is earnestly solicited.

Claims 12-16 depend from claim 11 and add further distinguishing elements. Claims 12-16 are thus also submitted to be allowable. Withdrawal of the rejection of claims 12-16 is earnestly solicited.

Claim 17:

Claim 17 recites:

The task switcher switching from execution of the second task to execution of a third task, said memory having a storage location for storing a return address corresponding to the third task, and an input for receiving information indicative of instructions of a task switching function that has been called by the second task.

Fleck neither teaches, discloses, nor suggests "the task switcher switching from execution of the second task to execution of a third task, said memory having a storage location for storing a return address corresponding to the third task, and an input for receiving information indicative of instructions of a task switching function that has been called by the second task," as discussed above with respect to the rejection of claim 1. Claim 17 is thus submitted to be allowable. Withdrawal of the rejection of claim 17 is earnestly solicited.

Rejection Under 35 U.S.C. § 103:

Claims 18 was rejected under 35 U.S.C. § 103(a) as unpatentable over Fleck. Claim 18 depends from claim 17 and adds further distinguishing elements. Fleck neither teaches, discloses, nor suggests "the task switcher switching from execution of the second task to execution of a third task, said memory having a storage location for storing a return address corresponding to the third task, and an input for receiving information indicative of instructions of a task switching function that has been called by the second task," as discussed above with respect to the rejection of claim 17. Claim 18 is thus also submitted to be allowable. Withdrawal of the rejection of claim 18 is earnestly solicited.

Claims 9 and 20:

Claims 9 and 20 were rejected under 35 U.S.C. § 103(a) as unpatentable over Fleck in view of "Applicant's Admitted Prior Art." Claims 9 and 20 depend from claims 1 and 17, respectively and add further distinguishing elements. Fleck neither teaches, discloses, nor suggests "the task switching function executing a return operation to the third data processing task," as discussed above with respect to the rejections of claim 1. Fleck neither teaches, discloses, nor suggests "the task switcher switching from execution of the second task to execution of a third task, said memory having a storage location for storing a return address corresponding to the third task, and an input for receiving information indicative of instructions of a task switching function that has been called by the second task," as discussed above with respect to the rejection of claim 17. The parts of the Application labeled "Prior Art" do not either, and thus cannot make up for the deficiencies of Fleck with respect to claims 9 and 20. Claims 9 and 20 are thus submitted to be allowable. Withdrawal of the rejection of claims 9 and 20 is earnestly solicited.

Conclusion:

Claims 1-9, and 11-20 are submitted to be allowable over the cited references. There being no further outstanding objections or rejections, it is submitted that the application is in condition for allowance. An early action to that effect is courteously solicited.

Finally, if there are any formal matters remaining after this response, the Examiner is requested to telephone the undersigned to attend to these matters.

If there are any additional fees associated with filing of this Amendment, please charge the same to our Deposit Account No. 19-3935.

Respectfully submitted,

STAAS & HALSEY LLP

By: 

Thomas E. McKiernan
Registration No. 37,889

Date: 09/10/07

1201 New York Ave, N.W., 7th Floor
Washington, D.C. 20005
Telephone: (202) 434-1500
Facsimile: (202) 434-1501